

Mod 10 Lab: Priority Queues

Implement the Priority Queue ADT with two (non-heap) data structures:

- PQ_UL - priority queue ADT with unordered list data structure
- PQ_OL - priority queue ADT with ordered (sorted) list data structure

Part 1 - class Entry

Your Priority Queues can both use the same Entry class.

Magic Methods

- `init` - gives entries (at least) two attributes:
 - `item`
 - `priority`
- `lt(self, other)` - return `True` if `self` has a lower priority than `other`, `False` otherwise.
- `eq(self, other)` - returns `True` if the two entries have the same priority *and* item.

Part 2 - Priority Queue classes PQ_UL and PQ_OL

Both priority queues should support the following ADT:

Magic Methods

- `len`
 - returns the number of entries in the priority queue

Non-magic Methods

- `insert(item, priority)`
 - adds `item` with given `priority` to priority queue
- `find_min()`
 - returns (but does not remove) the `item` with minimum `priority`.
- `remove_min()`
 - returns and removes the `item` with minimum `priority`. This means an item with priority 0 will be returned before an item with priority 5, for instance.

Special Cases/Notes

- Feel free to use the `lst.sort()` method in this assignment. It's $O(n \log n)$.

Submitting

At a minimum, submit the following files:

- `lab10.py`
 - contains classes
 - * PQ_UL - unordered list
 - * PQ_OL - ordered list

Students must submit **individually** by the due date (typically Sunday at 11:59 pm EST) to receive credit.

Grading

This assignment is 100% auto-graded.

Feedback

If you have any feedback on this assignment, please leave it [here](#).